

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-336641

(43)公開日 平成10年(1998)12月18日

(51) Int.Cl.⁶

H04N 7/24

識別記号

F I

H04N 7/13

$$\mathbf{z}$$

審査請求 未請求 請求項の数28 O L (全 20 頁)

(21)出願番号 特願平9-144306

(22) 出願日 平成9年(1997)6月2日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 五十崎 正明

東京都品川区北品川6丁目7番35号 ソニ
一株式会社内

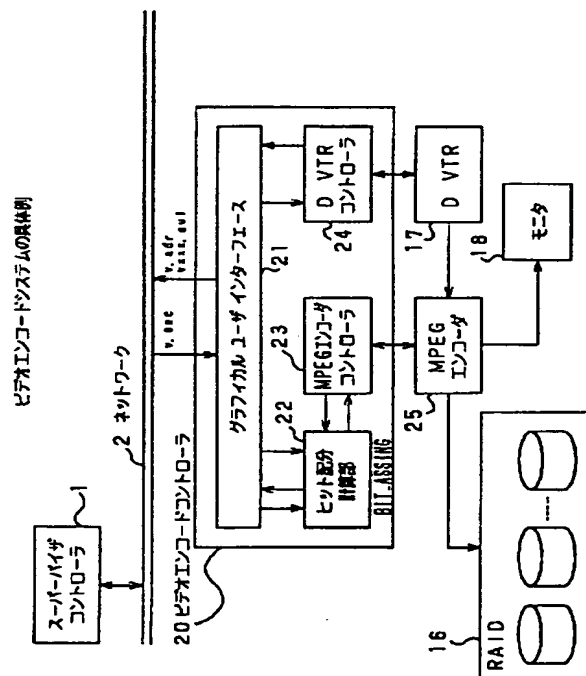
(74)代理人 弁理士 小池 晃 (外2名)

(54) 【発明の名称】 画像符号化方法及び装置並びに記録媒体並びに画像伝送方法

(57) 【要約】

【課題】 暗い部分での人の視覚感度は高いため、暗い動画でのノイズは知覚されやすく、暗いフェードイン／アウト点でのビットレートの大きさが十分でないとノイズが目立ってしまい問題となってくる。

【解決手段】 ビット配分計算処理部２２は、ＭＰＥＧエンコーダコントローラ２３の制御により、ビデオ素材のフレーム内平均輝度のレベルと時間変化量から、暗い部分を介して明暗が変化する区間、例えばフェードイン／フェードアウト区間を検出し、これらの区間の画像の難しさを表す符号化難易度の値に重み係数を乗じて符号化難易度を補正し、この補正した符号化難易度に応じて全体のビットレートを割り当てる。



【特許請求の範囲】

【請求項 1】 画像フレームの符号化難易度に応じて割り当てビット量を決定する画像符号化方法において、暗い部分を介して明暗が変化する区間を上記フレーム内の平均輝度のレベルの時間変化に基づいて検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定することを特徴とする画像符号化方法。

【請求項 2】 上記明暗が変化する区間では、上記符号化難易度に重み係数を乗算して補正した符号化難易度を算出することを特徴とする請求項 1 記載の画像符号化方法。

【請求項 3】 上記明暗が変化する区間は、複数の画像フレームからなる画像ブロックの数個分のゆっくりとした時間であることを特徴とする請求項 2 記載の画像符号化方法。

【請求項 4】 画像フレームの符号化難易度に応じて割り当てビット量を決定する画像符号化装置において、暗い部分を介して明暗が変化する区間を上記フレーム内の平均輝度のレベルの時間変化に基づいて検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を演算する演算手段を備えることを特徴とする画像符号化装置。

【請求項 5】 上記演算手段は、上記明暗が変化する区間では、上記符号化難易度に重み係数を乗算して補正した符号化難易度を算出することを特徴とする請求項 4 記載の画像符号化装置。

【請求項 6】 上記明暗が変化する区間は、複数の画像フレームからなる画像ブロックの数個分のゆっくりとした時間であることを特徴とする請求項 4 記載の画像符号化装置。

【請求項 7】 画像フレームの符号化難易度に応じて割り当てられたビット量の画像を記録している記録媒体において、暗い部分を介して明暗が変化する区間を上記フレーム内の平均輝度のレベルの変化に基づいて検出し、上記明暗が変化する区間における上記符号化難易度を補正し、上記補正した符号化難易度に応じて上記割り当てビット量を決定していることを特徴とする記録媒体。

【請求項 8】 画像フレームの符号化難易度に応じて割り当てビット量を決定して、符号化画像を伝送する画像伝送方法において、暗い部分を介して明暗が変化する区間を上記フレーム内の平均輝度のレベルの時間変化に基づいて検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定することを特徴とする画像伝送方法。

【請求項 9】 画像フレームの符号化難易度に応じて割り当てビット量を決定する画像符号化方法において、少なくとも一つのフレーム内符号化画像と、フレーム間

順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化に基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定することを特徴とする画像符号化方法。

【請求項 10】 上記明暗が変化する区間は、上記符号化難易度に重み係数を乗算して補正した符号化難易度を算出することを特徴とする請求項 9 記載の画像符号化方法。

【請求項 11】 上記重み係数の大きさを、対象となる上記画像符号化グループ単位の符号化難易度の和と、全体の画像符号化グループでの符号化難易度の和の平均値との比を基にして算出することを特徴とする請求項 9 記載の画像符号化方法。

【請求項 12】 上記明暗が変化する区間は、上記画像符号化グループの時間長さの数倍程度の時間であることを特徴とする請求項 9 記載の画像符号化方法。

【請求項 13】 画像フレームの符号化難易度に応じて割り当てビット量を決定する画像符号化装置において、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化に基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を演算する演算手段を備えることを特徴とする画像符号化装置。

【請求項 14】 上記演算手段は、上記明暗が変化する区間では、上記符号化難易度に重み係数を乗算して補正した符号化難易度を演算することを特徴とする請求項 13 記載の画像符号化装置。

【請求項 15】 上記演算手段は、上記重み係数の大きさを、対象となる上記画像符号化グループ単位の符号化難易度の和と、全体の画像符号化グループでの符号化難易度の和の平均値との比を基にして演算することを特徴とする請求項 13 記載の画像符号化装置。

【請求項 16】 上記明暗が変化する区間の時間長さは、上記画像符号化グループの時間長さの数倍程度の時間であることを特徴とする請求項 13 記載の画像符号化装置。

【請求項 17】 画像フレームの符号化難易度に応じて割り当てられたビット量の画像を記録している記録媒体において、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化に基づいて、暗い部分を介して明暗が変化する区間を検出し、上記明暗が変化する区間における上記符号化難易度を補正し、上記補正した符号化難易度に応じて

上記割り当てビット量を決定していることを特徴とする記録媒体。

【請求項 1 8】 画像フレームの符号化難易度に応じて割り当てビット量を決定して、符号化画像を送送する画像伝送方法において、

少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化に基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定することを特徴とする画像伝送方法。

【請求項 1 9】 画像フレームの符号化難易度に応じて割り当てビット量を決定する画像符号化方法において、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化及び特定の符号化画像の符号化難易度の比率の時間変化とに基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定することを特徴とする画像符号化方法。

【請求項 2 0】 上記明暗が変化する区間は、上記符号化難易度に重み係数を乗算して補正した符号化難易度を算出することを特徴とする請求項 1 9 記載の画像符号化方法。

【請求項 2 1】 上記重み係数の大きさを、対象となる上記画像符号化グループ単位の符号化難易度の和と、全体の画像符号化グループでの符号化難易度の和の平均値との比を基にして算出することを特徴とする請求項 1 9 記載の画像符号化方法。

【請求項 2 2】 上記明暗が変化する区間は、上記画像符号化グループの時間長さの数倍程度より短い時間であることを特徴とする請求項 1 9 記載の画像符号化方法。

【請求項 2 3】 画像フレームの符号化難易度に応じて割り当てビット量を決定する画像符号化装置において、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化及び特定の符号化画像の符号化難易度の比率の時間変化とに基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を演算する演算手段を備えることを特徴とする画像符号化装置。

【請求項 2 4】 上記演算手段は、上記明暗が変化する区間では、上記符号化難易度に重み係数を乗算して補正した符号化難易度を演算することを特徴とする請求項 2 3 記載の画像符号化装置。

【請求項 2 5】 上記演算手段は、上記重み係数の大き

さを、対象となる上記画像符号化グループ単位の符号化難易度の和と、全体の画像符号化グループでの符号化難易度の和の平均値との比を基にして演算することを特徴とする請求項 2 3 記載の画像符号化装置。

【請求項 2 6】 上記明暗が変化する区間の時間長さは、上記画像符号化グループの時間長さの数倍程度より短い時間であることを特徴とする請求項 2 3 記載の画像符号化装置。

【請求項 2 7】 画像フレームの符号化難易度に応じて割り当てられたビット量の画像を記録している記録媒体において、

少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化及び特定の符号化画像の符号化難易度の比率の時間変化とに基づいて、暗い部分を介して明暗が変化する区間を検出し、上記明暗が変化する区間における上記符号化難易度を補正し、上記補正した符号化難易度に応じて上記割り当てビット量を決定していることを特徴とする記録媒体。

【請求項 2 8】 画像フレームの符号化難易度に応じて割り当てビット量を決定して、符号化画像を送送する画像伝送方法において、

少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化及び特定の符号化画像の符号化難易度の比率の時間変化とに基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定することを特徴とする画像伝送方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】 本発明は、画像フレームの符号化難易度に応じて画像を符号化する画像符号化方法及び装置並びに記録媒体並びに画像伝送方法に関する。

【0 0 0 2】

【従来の技術】 ビデオ情報をディジタルビデオディスク (Digital Video Disk : DVD) やビデオ CD のようなパッケージメディアに蓄積する際、上記ビデオ情報に圧縮符号化処理を施すエンコードシステムでは、最初に素材の画像の符号化難易度 (Difficulty) を測定し、その符号化難易度を元に、パッケージメディアの記録容量内の与えられたバイト数に収まるように、各ビデオ情報のフレームごとにビット配分 (以下、Bit assign) 処理を行ってエンコードするという方法が一般に採用されている。以下、このエンコード方法を 2 パスエンコーディング方法という。

【0 0 0 3】 例えば、上記ディジタルビデオディスク用に、上記 2 パスエンコーディング方法を採用して、ビデ

オ情報を圧縮符号化するビデオエンコードシステムの具体例を図16に示す。

【0004】図16において、ビデオエンコードの制御を行うビデオエンコードコントローラ10は、システム全体を管理するスーパーバイザコントローラ1に、ネットワーク2を介して接続されている。

【0005】スーパーバイザコントローラ1はオペレーティングシステムを構成するプログラムの内、特にシステム全体の動きを監視し、効率的に制御するプログラムであるスーパーバイザを実行するコントローラである。このビデオエンコードシステムにおいてはDVDのオーサリングシステム全体の管理を行い、ビデオ、オーディオ、字幕やメニューといった各エンコードシステムにエンコード条件を与え、エンコード結果の報告を受ける。

【0006】このビデオエンコードシステムの具体例に対しては、例えばv.enc というファイルによってビデオエンコード条件を指定している。そして、ビデオエンコードコントローラ10側からは、RAID16 (Redundant Arrays of Inexpensive Disks) 上に書き込まれたアドレスv.adrと、エンコード結果のビットストリームがオーディオや字幕、メニュー等のサブピクチャとマルチプレックスされる際に必要とされるデータ(vxxx.aui)を報告している。ここで、RAID16は、エンコード結果のビットストリームを記録する大容量記録媒体であり、ハードディスクドライブ(HDD)等を複数並列に接続して記録容量と転送速度性能を向上させている。

【0007】ビデオエンコードコントローラ10は、グラフィカルユーザインターフェース(Graphical User Interface: GUI)11と、後述するビット配分計算処理プログラム(Bit_Assign)を格納しているビット配分計算部12と、このビット配分計算部12内部のビット配分計算処理プログラム(Bit_Assign)を実行するMPEGエンコーダコントローラ13と、デジタルVTRコントローラ14とを備えている。

【0008】ユーザは、グラフィカルユーザインターフェース11を用い、ビット配分計算部12の上記ビット配分計算処理プログラム(BIT_ASSIGN)と、MPEGエンコーダコントローラ13の3つのプログラムを管理することができる。また、DVTRコントローラ14も管理できる。

【0009】MPEGエンコーダコントローラ13は、上記ビット配分計算部12内部の上記ビット配分計算処理プログラム(BIT_ASSIGN)を実行すると共に、MPEGエンコーダ15を制御する。また、DVTRコントローラ14はDVTR17を制御する。このDVTR17はMPEGエンコーダ15に接続しており、MPEGエンコーダ15はエンコードした結果を表示するためにモニタ18に接続している。さらに、MPEGエンコーダ15は、エンコード結果を記録するために上記RAID16にも接続している。

【0010】MPEGエンコーダ15では、動き補償予測による時間方向の冗長度の除去を行って、ビデオ情報を圧縮している。また、MPEGエンコーダ15では、フレーム内だけで符号化されるフレーム内符号化画像をIピクチャ(Intra Coded)、過去の画面から現在を予測することによって符号化されるフレーム間順方向予測符号化画像をPピクチャ(Predictive Coded)、過去、未来の両方向の画像から現在を予測することによって符号化される双方向予測符号化画像をBピクチャ(Bidirectionally Predictive Coded)として用い、ビデオ情報を圧縮符号化している。ここでは、必ずIピクチャを1つ含むピクチャーのまとまりを図17に示すようなGOP(Group of Pictures)としている。この図17において、GOPのフレーム数Nは15であり、表示順のGOPの先頭は、Iピクチャの前で、P又はIピクチャの次のBピクチャーである。GOPの最後は、次のIピクチャーの前の最初のPピクチャーである。

【0011】このビデオエンコードシステムの動作について図18のフローチャートを参照して説明する。先ず、ステップS1で、スーパーバイザコントローラ1からネットワーク2経由でビデオに割り当てるビット総量や最大レートなどのエンコード条件v.encが与えられ、MPEGエンコーダコントローラ13はエンコード条件を設定する。その後、ステップS2でMPEGエンコーダコントローラ13の制御の基にMPEGエンコーダ15がエンコード素材の符号化難易度を測定する。ここでは、各画素のDC値や動きベクトル量MEも読んでおく。そして、これらの測定結果により、ファイルを作成しておく。

【0012】実際の符号化難易度の測定は以下のように行う。エンコード素材となるビデオ情報はDVTR17によってマスターテープであるデジタルビデオカセットから再生される。MPEGエンコードコントローラ13は、MPEGエンコーダ15を介して、DVTR17によって再生されたビデオ情報の符号化難易度を測定する。

【0013】ここでは、符号化の際に量子化ステップ数を固定値に設定した条件で発生ビット量を測定する。動きが多く、高い周波数成分が大きい画像では発生ビット量が大きくなり、静止画や平坦な部分が多い画像では発生ビット量が少なくなる。この発生ビット量の大きさを上記符号化難易度としている。

【0014】次に、ステップS3では、ステップS1で設定されたエンコード条件を元に、ステップS2で測定された各ピクチャーの符号化難易度の大きさに応じて、MPEGエンコードコントローラ13がビット配分計算部12内部のビット配分計算プログラム(BIT_ASSIGN)を実行し、割り当てビット量(ターゲット量:target)の配分計算を行う。

【0015】そして、このステップS3でのビット配分

計算による結果を使ってエンコードを実行するかどうかをMPEGエンコーダ15に内蔵されているローカルデコーダ出力の画質によってユーザに判断させる。

【0016】実際には、ステップS4で、上記ビット配分によるビットストリームをRAID16に出力しないで、任意の処理範囲を指定できるプレビューモード (Preview) を行って、ユーザが画質をチェックする。

【0017】ステップS5の画質評価で画質に問題がない場合にはステップS6に進み、MPEGエンコーダ15によるエンコード処理を実行するが、画質に問題がある場合には、ステップS8に進み、問題のある部分のレートを上げるとか、フィルターレベルを調整するといった画質調整のためのカスタマイズ作業を行ってから、ステップS9で再びビット配分計算部12内部のビット配分計算処理プログラム (BIT_ASSIGN) を実行し、ビット再配分計算を行う。

【0018】その後、ステップS4に戻り、カスタマイズした部分をプレビューして、ステップS5で画質を確認し、すべての部分が良ければステップS6に進み、全体のエンコードをMPEGエンコーダ15に実行させる。エンコード結果であるビットストリームは、ステップS7でSCSI (Small Computer System Interface) 経由で直接、RAID16に書き込まれる。

【0019】ステップS6でのエンコード後、ビデオエンコードコントローラ10は上述したようなエンコード結果情報をネットワーク経由でスーパーバイザコントローラ1に報告する。

【0020】この図18のフローチャートにおいて、ステップS2、ステップS4及びステップS6を除いた各ステップの処理はオフライン処理を意味している。

【0021】以下、特に、ステップS3でMPEGエンコードコントローラ13によって実行されるビット配分計算部12内のビット配分計算処理について詳細に説明する。まず、スーパーバイザコントローラ1から上記ビット総量 (QTY_BYTES) と、最大ビットレート (MAXRATE) が指定されると、これに対して、MPEGエンコーダコントローラ13は、最大ビットレート (MAXRATE) 以下になるように制限を加えた総ビット数 (USB_BYTES) を求め、この値からGOPのヘッダ (GOP header) に必要なビット数 (TOTAL_HEADER) を引いた値と、全体のフレーム総数からターゲット数の総和の目標値となるSUPPLY_BYTESを算出する。

【0022】そして、このSUPPLY_BYTESの大きさに収まるようにビット配分計算部12内部のビット配分計算処理プログラムを実行し、各ピクチャーへの割り当てビット量 (以下、ターゲット量: target) を配分する。

【0023】このステップS3でのビット配分計算処理を詳細に示したのが図19のフローチャートである。

【0024】まず、ステップS11で上述したように、スーパーバイザコントローラ1から送られた上記ビット

総量 (QTY_BYTES) と、最大ビットレート (MAXRATE) が入力されると、MPEGエンコーダコントローラ13は、上述したようにSUPPLY_BYTESを算出する。

【0025】次に、上記図18のステップS2の符号化難易度の測定で作成された測定ファイルをステップS12でそのまま読み込み、符号化難易度の測定の際に、併せて測定された各画像のDC値や動きベクトル量MEの大きさのパラメータの変化量から、ステップS13でシーンが変化するポイントを見つける。

【0026】このステップS13でのシーンチェンジ検出/処理は、本件出願人が既に特願平8-274094号明細書及び図面にて開示した「映像信号処理装置」に応じてシーンチェンジ点を検出する処理である。

【0027】この「映像信号処理装置」は、映像信号の各フレームの直流レベルを検出し、この直流レベルを曲線近似して得られる誤差値より、上記映像信号のシーンチェンジのフレームを検出して、シーンチェンジ点を明らかにする。

【0028】そして、図20に示すように、シーンがチェンジしたとして検出したポイントは、PピクチャーをIピクチャーに変更して、画質改善を計る。

【0029】次に、ステップS14でチャプター (CHAPTER) 境界処理を行う。DVD再生装置でのチャプターサーチ時には、特定されないピクチャーからジャンプしてくることになるが、その場合でも再生画像の乱れがないようにするため、図17に示すようにチャプターの位置が必ずGOPの先頭になるようにピクチャータイプを変更する。図21ではPピクチャーをIピクチャーに変更している。

【0030】このようなステップS12、ステップS13での一連の作業の結果、ピクチャータイプ (I、P、Bピクチャー) の変更処理が実行されると、符号化難易度測定時のピクチャータイプが変更されるため、ステップS15で変更後のピクチャータイプに合わせた符号化難易度の値に補間/補正する。

【0031】ステップS15での符号化難易度の補間/補正によって得られた符号化難易度と、全体に与えられたビット数 (SUPPLY_BYTES) に応じて、ステップS16で各ピクチャーごとのターゲットビット数を計算する。

【0032】そして、ステップS17でRAID16にエンコード結果のビットストリームを書き込む際のアドレスの計算を行った上で、ステップS18に進み、エンコード用のコントロールファイルを作成する。

【0033】例えば、ビット配分計算の具体例として、まずGOP単位にビット量を配分してから、各GOP内で各ピクチャーの符号化難易度に応じたビット配分を行う場合を説明する。ここでは、各GOP毎の符号化難易度の和であるGOP_DIFFに応じて、エンコードする際のGOP単位のビット割り当て量 (GOP_TARGET) を配分している。このGOP_DIFFとGOP_TARGETとを変換するもっとも

簡単な関数の例を図22に図示する。

【0034】この例では、縦軸YをGOP_TARGET、横軸XをGOP_DIFFとして、 $Y=AX+B$ という評価関数を用いている。なお、全てのピクチャーの符号化難易度の総和(DIFFICULTY_SUM)を算出しておく。

$$USB_BYTES = \min(QTY_BYTES, MAXRATE \times KT \times total_frame_number)$$

・・・(1)

のように求める。

【0036】ここで、NTSCの場合 $KT=1/8(\text{bits})/30(\text{Hz})$ 、PALの場合 $1/8(\text{bits})/25(\text{Hz})$ である。また、total_frame_number はエンコードする素材のフレーム総数、mi

$$SUPPLY_BYTES = USB_BYTES - TOTAL_HEADER$$

のように求める。

【0038】次に、全てのピクチャーの符号化難易度の総和DIFFICULTY_SUM = $\sum \text{difficulty}$

と表せる。

【0039】また、GOP_TARGETの最小値を次の(4)式 $B = GOP_MINBYTES$

とする。

【0040】すると、図22に示した評価関数のような $\sum y = A \times \sum x + B \times n$ が得られる。

【0041】ここで、 $\sum y = SUPPLY_BYTES$ 、 $\sum x = DIFFICULTY_SUM$ 、 $GOP_TARGET = A \times GOP_DIFF + B$

と表せる。

【0043】その後、各GOP内で各ピクチャーの符号化難易度に応じたビット配分を行う。GOP内での各ピクチャーの配分を符号化難易度の大きさに比例させた

$$target(k) = GOP_TARGET \times difficulty(k) / DIFFICULTY_SUM \quad \dots (6)$$

($1 \leq k \leq \text{GOP 内の picture 数}$)

この場合、素材の中に極端に難しい(GOP_DIFFが大きい)ピクチャーがあると、非常に大きいGOP_TARGET量となってしまう、システムで許容されている最大レートを越えてしまうため、GOP_MAXBYTESといった固定量でリミットをかけることが必要である。また、最小のターゲット量もGOP_MINBYTESで制限する。具体的には、以下に説明するアルゴリズムによっている。

【0045】MPEGビデオのエンコード時には、仮想 $OCCUPANCY_UP(0) = VBVMAX \times 2/3$

以下のOCCUPANCY_UPはグラフ上の各ピクチャーの上側のポイント、OCCUPANCY_DOWNはグラフ上の各ピクチャーの下側のポイントを意味している。

【0048】DVDのバッファサイズVBVMAX(1.75Mbit/s)に対して、k番目のピクチャーのバッファのスタート点をOccupancy_up(k)、k番目のピクチャーのターゲット量をtarget(k)とすると、ピクチャーにビットを吐

$$OCCUPANCY_DOWN(k) = OCCUPANCY_UP(k) - target(k) \quad \dots (8)$$

$$OCCUPANCY_UP(k+1) = OCCUPANCY_DOWN(k) + SYSTEM_SUPPLY \quad \dots (9)$$

この供給後のバッファ残量は、図23の図中の右上に上る量に相当する。供給されるビットレートが大きいほ

【0035】まず、最大ビットレート以下になるように制限を加えた総ビット数USB_BYTESを、スーパーバイザコントローラ1から与えられたビット総量QTY_BYTESと、最大ビットレートMAXRATEを使って、

n(s, t)はs, tの内で小さい方を選択する関数である。

【0037】また、SUPPLY_BYTESは、上記(1)式で求めたUSB_BYTESからGOPのヘッダに必要なビット数TOTAL_HEADERを引いて、

・・・(2)

和は、

・・・(3)

のように、

・・・(4)

DIFFICULTY_SUM、nはGOPの総数である。

【0042】よって $A = (SUPPLY_BYTES - B \times n) / DIFFICULTY_SUM$ となる。すると、各GOP毎のターゲット量は、

・・・(5)

場合には、各ピクチャーのターゲット量は以下の(6)式で求められる。

【0044】

・・・(6)

デコーダのバッファ残量を考慮しながらビット配分することが義務付けられている。この仮想バッファ残量の計算をVBV(Video buffering Verifier)という。

【0046】まず、図23を用いてVBV計算方法を説明する。この計算の最初のOCCUPANCY_UP(0)は次の(7)式に示すように、固定値(この例ではVBVMAX * 2/3)からスタートする。

【0047】

・・・(7)

き出したあとのバッファ残量OCCUPANCY_DOWN(k)は後述する(8)式で表される。このバッファには、デコーダのピックアップからビデオのデータ量に応じたビットレートのデータ量(SYSTEM_SUPPLY)が蓄積される。この供給後のバッファ残量OCCUPANCY_UP(k+1)は次の(9)式で表される。

【0049】

・・・(8)

・・・(9)

ど傾きは大きくなり、バッファにデータがたまりやすくなる。バッファがいっぱいになった場合には、ピック

アップからバッファへの供給がストップするため、バッファのオーバーフローに関しては考慮する必要はない。このことは、ある設定値ちょうどに制御する必要はなく、設定値以上になるように制御すれば良いことを意味している。

$$\text{SYSTEM_SUPPLY} = \text{MAXRATE}(\text{bps}) * \text{KT}$$

のように求める。

【0051】図24にGOP単位でのターゲットビット配分計算をおこなった例を示す。図24の(A)は評価関数とGOP_MAXRATE制限を考慮して求めたターゲット量に対して上記VBVバッファ計算をおこなった場合である。ここで、図24の(A)での[1]、[4]、

「7」のピクチャーでVBVバッファの下限であるVBVMINの値を下回っている。そこで、VBVがVBVMINを下回った

$$r = (\text{Occ_start} - \text{VBVMIN}) / (\text{VBVSTART} - \text{Occ_min}) \quad \dots (11)$$

各ターゲットに対して $\text{target}(j) = \text{target}(j) * r$ ($k\text{start} \leq j \leq k$) とする。

【0052】このようにして求められたターゲット量を用いて作成されたコントロールファイルによるエンコード処理を行うことで、素材の画像の難しさに応じた可変ビットレートエンコーディングが実行される。

【0053】ところで、編集されたビデオ素材では、だんだんと画面を暗くしていき、黒い画面から再びだんだんと画面を明るくして異なるシーンを接続するフェードアウト／フェードインの手法が良く用いられる。

【0054】図25は、暗い画面からフェードインする場合の説明図である。この内、図25の(a)には、輝度のフレーム単位での平均値DCの変化を示す。例えば、DCは0～255までの256段階で表され、大きい値ほど明るい状態を示し、小さいほど暗い状態を示す。図25の(b)には、仮エンコードによって測定されたk番目のフレームの符号化難易度(Difficulty)の値 $\text{gen_bit}(k)$ を示す。画像が難しいほどこの値 $\text{gen_bit}(k)$ が大きい。例えば、黒い静止画面では、画像が簡単なため、Iピクチャーの gen_bit は小さくなる。また、動きがないことと、フレーム相関が非常に大きいので、Pピクチャー、Bピクチャーの gen_bit も非常に小さい値となる。

【0055】図25の(a)に示したフェードインの間では時間的に前にあるIピクチャーより、後ろにあるP、Bピクチャーは、DCレベルが異なることからフレーム相関が少なくなるため、図25の(b)に示すように、P、Bピクチャーの gen_bit の値は相対的に大きくなる。

【0056】図25の(c)には、図25の(b)に示した gen_bit によって、I、B、Pピクチャーに配分されるターゲットビット量 $\text{target}(k)$ を示す。また、図25の

(d)には、図25の(c)のターゲットビット量 $\text{target}(k)$ に応じたフェードイン区間のビットレート配分を示す。

【0057】暗いシーンでは、通常の動画のシーンよりも画像の難しさが小さいため、符号化難易度は、平均的

【0050】逆に、各ピクチャーのデータ量が大きいと、バッファにたまったデータは減少する。このバッファ残量が一定値以下にならないようにターゲットビット量を計算する。そして、ビデオのデータ量に応じたビットレートのデータ量のSYSTEM_SUPPLYを、

$$\dots (10)$$

ピクチャーを含むGOPのターゲット量を削減させる。GOP内でVBV制限を加える前のターゲット量でVBV計算を実行したときのOCCUPANCYの最小値を Occ_min とすると調整量は以下の式であらわされる。ここで、制限をおこなうスタート点 $k\text{start}$ は、 $\text{OCCUPANCY_UP}(k)$ が基準値(VBVLIN:たとえば $\text{VBVMAX} * 3/4$)以上の k の値で、このときの $\text{OCCUPANCY_UP}(k)$ の値を Occ_start とする。

$\text{OCCUPANCY_MIN} < \text{VBVMIN}$ の時

な符号化難易度よりも低い値となる。全体のビットレートの配分は、符号化難易度の値によって決定されるので、このような暗いフェードイン／フェードアウト点でのビットレートは小さくなる。

【0058】

【発明が解決しようとする課題】しかし、暗い部分での人の視覚感度は高いため、暗い動画でのノイズは知覚されやすく、暗いフェードイン／アウト点でのビットレートの大きさが十分でないとノイズが目立ってしまい問題となってくる。

【0059】本発明は、上記実情に鑑みてなされたものであり、レート配分計算時に通常処理よりも多くレートを割り当て、フェードイン／フェードアウトポイントの画質を改善できる画像符号化方法及び装置並びに記録媒体並びに画像伝送方法の提供を目的とする。

【0060】

【課題を解決するための手段】本発明に係る画像符号化方法及び装置は、上記課題を解決するために、暗い部分を介して明暗が変化する区間を上記フレーム内の平均輝度のレベルの時間変化に基づいて検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定する。このため、補正された配分されるレートは平均ビットレートの近傍の値となる。

【0061】本発明に係る画像符号化方法及び装置は、上記課題を解決するために、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化に基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定する。

【0062】また、本発明に係る画像符号化方法及び装置は、上記課題を解決するために、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画

像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化及び特定の符号化画像の符号化難易度の比率の時間変化とに基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定する。

【0063】また、本発明に係る記録媒体は、上記課題を解決するために、暗い部分を介して明暗が変化する区間を上記フレーム内の平均輝度のレベルの変化に基づいて検出し、上記明暗が変化する区間における上記符号化難易度を補正し、上記補正した符号化難易度に応じて上記割り当てビット量を決定した画像を記録している。

【0064】また、本発明に係る記録媒体は、上記課題を解決するために、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化に基づいて、暗い部分を介して明暗が変化する区間を検出し、上記明暗が変化する区間における上記符号化難易度を補正し、上記補正した符号化難易度に応じて上記割り当てビット量を決定した画像を記録している。

【0065】また、本発明に係る記録媒体は、上記課題を解決するために、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化及び特定の符号化画像の符号化難易度の比率の時間変化とに基づいて、暗い部分を介して明暗が変化する区間を検出し、上記明暗が変化する区間における上記符号化難易度を補正し、上記補正した符号化難易度に応じて上記割り当てビット量を決定した画像を記録している。

【0066】また、本発明に係る画像伝送方法は、上記課題を解決するために、暗い部分を介して明暗が変化する区間を上記フレーム内の平均輝度のレベルの時間変化に基づいて検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定する。

【0067】また、本発明に係る画像伝送方法は、上記課題を解決するために、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化に基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定する。

【0068】また、本発明に係る画像伝送方法は、上記課題を解決するために、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グルー

プ単位での平均輝度の時間変化及び特定の符号化画像の符号化難易度の比率の時間変化とに基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定する。

【0069】

【発明の実施の形態】以下、本発明に係る画像符号化方法及び装置の実施の形態について図面を参照しながら説明する。

【0070】この実施の形態は、例えばデジタルビデオカセットテープに記録されたビデオ素材をデジタルビデオディスク(Digital Video Disk: DVD)用に、2パスエンコーディング方法を採用してエンコードするためのビデオエンコードシステムであり、図1に示すような構成である。

【0071】このビデオエンコードシステムは、上記図16に示したビデオエンコードシステムと基本的に構成を同じにしているが、ビデオエンコードコントローラ20内部でのビット配分計算処理動作を従来と異ならせている。

【0072】この図1に示したビデオエンコードシステムは、ビデオ素材のフレーム内平均輝度のレベルと時間変化量から、暗い部分を介して明暗が変化する区間、例えばフェードイン/フェードアウト区間を検出し、これらの区間の画像の難しさを表す符号化難易度の値に重み係数を乗じて符号化難易度を補正し、この補正した符号化難易度に応じて全体のビットレートを割り当てる画像符号化方法を適用したプログラムに従って、図2に示すようなビット配分計算処理を行う。このビット配分計算処理は、MPEGエンコーダコントローラ23によって制御されて、ビット配分計算部22で実行される。

【0073】このビット配分計算処理を図2のフローチャートにしたがって以下に説明する。

【0074】この図2に示すビット配分計算処理は、上記図19に示したフローチャートのステップS15とステップS16との間に、ステップS20としてフェードイン/フェードアウト点でのDifficultyの重み付け処理工程を設けている。

【0075】以下、このステップS20でのフェードイン/フェードアウト点でのDifficultyの重み付け処理工程を説明する。

【0076】このステップS20では、ビデオ素材である入力画像の平均輝度のレベルの時間変化、GOP内のIピクチャの符号化難易度Difficultyの割合によってフェードイン/フェードアウトの区間を検出し、この区間の符号化難易度に重み係数を乗じて符号化難易度を補正している。

【0077】そして、次のステップS16で、この補正した符号化難易度に応じて、ターゲットビット量を計算し、全体のビットレートを割り当ててビットレートが極

端に小さくならないように保護する。

【0078】このステップS20とステップS16での処理手順について暗い画面からフェードインする場合を具体例として図3を用いて説明する。

【0079】図3の(a)には、DC値の変化を示す。図3の(b)には、仮エンコードによって測定されたk番目のフレームの符号化難易度(Difficulty)の値の $gen_bit(k)$ を示す。図3の(c)には、重み付け係数 $rate_ctl$ を示す。図3の(d)には、上記重み付け係数 $rate_ctl$ によって補正されたk番目のフレームの符号化難易度の値 $gen_bit(k)$ を示す。図3の(e)には、図3の

(d)に示した補正後の $gen_bit(k)$ によって、I、B、Pピクチャに配分されるターゲットビット量 $target(k)$ を示す。また、図3の(f)には、図3の(e)のターゲットビット量 $target(k)$ に応じたフェードイン区間のビットレート配分を示す。

【0080】図2のステップS20で、フレーム内平均輝度のレベルと時間変化量から、図3の(a)に示したフェードイン区間を検出し、図3の(b)に示した画像の難しさを表す符号化難易度の値 $gen_bit(k)$ に、図3の(c)に示す重み係数 $rate_ctl$ を乗じて、符号化難易度値を補正して図3の(d)に示すような補正符号化難易度値 $gen_bit(k)$ を求める。ここでは、フェードイン区間の重み係数 $rate_ctl$ を初期値(1)より大きくして補正符号化難易度値 $gen_bit(k)$ を求めている。

【0081】そして、図2のステップS16で、図3の(d)に示した補正符号化難易度値 $gen_bit(k)$ を基にターゲット量を割り当ててやれば、図3の(f)に示すように平均ビットレートに十分近いビットレートを確保できる。

【0082】上記ステップS20のフェードイン/フェードアウト点でのdifficultyの重み付け処理工程は、図4に示すようなステップS21～ステップS24となる。

【0083】ステップS21では、GOP単位の各パラメータの測定や、重み係数の初期化を行う。後述するステップS22でのフェードイン点の検出や、ステップS23でのフェードアウト点の検出には、平均輝度としてGOP単位でのDC値の平均値 avr_dc と、GOP内のIピクチャの符号化難易度の割合としてBピクチャとIピクチャの符号化難易度の比率 b_rate を用いている。これら avr_dc や、 b_rate を求めるため、予めGOP単位の各パラメータを測定しておく必要がある。

【0084】例えば、 j ($1 \leq j \leq total_gopnb$) 番目のGOPのDC値の平均値を表す $avr_dc[j]$ は、 gop_dc をGOP単位のDC値の和とし、 gop_pcnt をGOP内のピクチャ数とすると、 $avr_dc = gop_dc / gop_pcnt$ で表せる。ここで、 $total_gopnb$ はGOPの総数を表す。

【0085】また、 j 番目のGOPのBピクチャとIピクチャの符号化難易度difficultyの比率を表す $b_rate[j]$

は、 b_bits をGOP内のBピクチャのdifficultyの値の和とし、 b_nb をGOP内のBピクチャの数とし、 i_bits をGOP内のIピクチャのdifficultyの値とすると、 $b_rate = b_bits / b_nb / i_bits$ で表せる。

【0086】また、仮エンコードによって測定されたdifficultyのGOP単位での和の平均値である後述する avr_gop_genbit は、 $gop_genbit[j]$ をj番目のGOPの仮エンコードによって測定されたdifficultyのGOP単位での和とすると、 $avr_gop_genbit = \sum gop_genbit / total_gopnb$ で表せる。

【0087】また、 j 番目のGOPのdifficultyの重み係数を表す $rate_ctl[j]$ は、初期値の際に1.0に設定される。

【0088】また、GOPのスタートを表す $gop_start[k]$ は、最初からk番目のフレームがGOPの先頭の場合“1”とし、それ以外は“0”とする。

【0089】次に、ステップS22ではフェードイン点の検出と重み係数 $rate_ctl[j]$ の算出を行い、ステップS23では、フェードアウト点の検出と重み係数 $rate_ctl[j]$ の算出を行う。

【0090】そして、ステップS24では、GOPの仮エンコードによって測定された gop_genbit に上記それぞれの重み係数 $rate_ctl[j]$ を乗算してdifficultyの重み付けを行い、GOPの補正符号化難易度値 gop_genbit を求める。

【0091】ステップS22のフェードイン点の検出、重み係数の算出処理工程や、ステップS23でのフェードアウト点の検出、重み係数の算出処理工程では、一般的にフェードイン、フェードアウトの速度が一定ではないので、例えばゆっくりと変化する場合と、通常の早さで変化する場合とで、各点の検出方法を異ならせる。

【0092】実際のビデオ素材の編集時には、例えば黒い画面からタイトルをゆっくり表示したり、夕陽をゆっくり沈ませて暗くするような、ゆっくりなだらかに暗い部分を介して明暗を変化させる場合と、場面を変えるのに短時間で明るい画面から暗い画面にし、さらに次の明るい画面に変化させるような通常に明暗を変化させる場合がある。

【0093】以下、このなだらかに暗い部分を介して明暗を変化させる場合と、通常に明暗を変化させる場合とに分けて、上記フェードイン区間及び上記フェードアウト区間でのフェードイン点及びフェードアウト点の検出と、重み係数の算出処理を説明する。

【0094】図5には、ステップS34になだらかに変化するフェードイン点の検出処理工程と、ステップS35に通常で変化するフェードイン点の検出処理工程とを併せ持つフェードイン点の検出処理のフローチャートを示す。

【0095】まず、ステップS31では、入力するビデオ素材の初期設定を $k=j=0$ 、 $lstart=j$ 、 $fmode=0$ として行

う。そして、ステップS32で、GOPの始まりを検出した時点から、ステップS34のなだらかに変化するフェードイン点の検出処理工程と、ステップS35の通常に変化するフェードイン点の検出処理工程を実行する。そして、これらの検出処理工程を、ステップS36及びステップS37での処理及び判定を介して繰り返す。

【0096】上記ステップS34に示したなだらかなフェードイン点の検出処理工程を、サブルーチンを示す図6とタイミングチャートを示す図7を用いて説明する。

【0097】ここでは、例えば、DC_LOW=25、DC_HIGH=70、第1のしきい値DELTA1=-5、第2のしきい値DELTA2=10、定数k1=0.8、定数k2=1.0、RATE_UP=3.0、BR_DELTA=50と設定する。

【0098】図7のタイミングチャートでは、(a)が平均輝度のレベルDCの変化を示す。また(b)がGOPのDC値の平均値AVR_DCの変化を示す。この(b)にはDC_LOWと、DC_HIGHとが破線で記されている。また、

(c)がGOPのBピクチャとIピクチャの比率b_rate(図中B/Iと記す)の変化を示す。また、(d)がGOP単位のモードfmodeの変化を示す。

【0099】このようなゆっくりと立ち上がるフェードインの区間のGOPでは、隣接するGOPとの画像の差が小さいため、上記図7の(c)に示すようなb_rateの大きさも急激には変化しない。そこで、このような場合には、上記図7の(c)に示すようなb_rateを用いず、図7の(b)に示すようなavr_dcの値の変化だけで上記フェードイン区間を検出する。

【0100】まず、図6のステップS41で、連続する2つのGOPのavr_dcの値が、DC_LOW以下であるか否かを判定し、共に以下であればステップS42に進みfmode=1とし、この位置のGOP番号(j)をフェードイン点fstart=jとする。

【0101】また、ステップS43では1つ前のGOPのavr_dcとの変化量が予め設定されている第1のしきい値DELTA1以下であるか否かを判定し、以下であればステップS42に進みfmode=1とし、この位置のGOP番号(j)をフェードイン点fstart=jとする。

【0102】また、ステップS44では1つ前のGOPのavr_dcとの変化量が上記第1のしきい値DELTA1と第2のしきい値DELTA2との間にあるか否かを判定し、間にあればステップS45に進みfmodeを2とする。

【0103】そうでない場合には、fmode=1とし、その位置のGOP番号(j)に対してfstart=jとする。

【0104】ステップS46及びステップS47を通して、fmodeが2の時に、avr_dcがDC_HIGH以上になったか否かを判定し、以上になればステップS48及びステップS49を通して、fstartからその位置までのGOPの重み係数rate_ctl[i]をavr_gop_genbit/gop_genbit[i]*k1として求める。重み係数rate_ctl[i]の大きさはステップS50～ステップS53を介して1.0～RATE_UPの範

囲に制限される。

【0105】そして、これらの処理は、ステップS54及びステップS55を通して繰り返され、j番目までいったら、ステップS56でfmode=1として、図5のステップS35に進む。

【0106】この図6に示したゆっくりしたフェードイン点の検出処理では、重み係数rate_ctlは、初期設定値を1としたときに、ステップS49に示したように、GOP単位のdifficultyと、平均値との比率に定数k1を乗じた値となる。

【0107】もし、重み係数の値を2倍といった固定値に設定すると、補正されたdifficultyが大きくなりすぎ、不必要な大きさまでビットレートが配分される可能性がある。また、全体のdifficultyの分布によっては、2倍では不十分なこともあり得る。

【0108】そこで、本発明の画像符号化方法によって重み係数を決定することで、補正されて配分されたレートは平均ビットレートの近傍の値になるため、過小、過度にレートが補正されることがなくなる。

【0109】次に、上記図5のステップS35に示した通常に変化するフェードイン点の検出処理工程を、サブルーチンを示す図8とタイミングチャートを示す図9を用いて説明する。

【0110】ここでも、例えば、DC_LOW=25、DC_HIGH=70、第1のしきい値DELTA1=-5、第2のしきい値DELTA2=10、定数k1=0.8、定数k2=1.0、RATE_UP=3.0、BR_DELTA=50と設定する。

【0111】図9のタイミングチャートでも、(a)が平均輝度のレベルDCの変化を示す。また(b)がGOPのDC値の平均値AVR_DCの変化を示す。この(b)にはDC_LOWと、DC_HIGHとが破線で記されている。また、

(c)がGOPのBピクチャとIピクチャの比率b_rate(図中B/Iと記す)の変化を示す。

【0112】このような通常で立ち上がるフェードイン区間のGOPでは、隣接するGOPとの画像の差が大きい。図9の(b)に示したavr_dcがDC_LOWを下回るような、黒い静止画面では、動きがないことと、フレーム相関が非常に大きいため、P、Bピクチャのgen_bitも非常に小さくなることから図9の(c)に示すようにb_rateの値が非常に小さくなる。また、フェードインの間では、時間的に前にあるIピクチャと、後ろにあるP、BピクチャとのDCレベルが異なることからフレーム相関が少なくなり、P、Bピクチャのgen_bitの値は相対的に大きくなる。よって、b_rateの値が非常に大きくなる。このため、図9の(c)に示すb_rateの変化量と、図9の(b)に示すavr_dcの値でフェードイン区間を決定することができる。

【0113】まず、図8のステップS61で、現在のGOPのb_rateと1つ前のGOPのb_rateの差がBRのしきい値BR_DELTAより大きく、かつ1つ前のGOPのavr_dc

の値がDC_LOW以下であるか否かを判定し、YESのときにはその位置からのGOPの区間の重み係数rate_ctlを、ステップS62を介したステップS63で、 $rate_ctl[i] = avr_gop_genbit / gop_genbit[i] * k2$ として求める。

【0114】重み係数の大きさは、ステップS64～ステップS67を通して、1.0～RATE_UPの範囲に制限される。そして、ステップS68及びステップS69の処理及び判定を通して、上記重み係数をフェードイン点から2個後ろまでのGOPの区間で計算する。

【0115】次に、図10には、ステップS74になだらかに変化するフェードアウト点の検出処理工程と、ステップS75に通常の変化するフェードアウト点の検出処理工程とを併せ持つフェードアウト点の検出処理のフローチャートを示す。

【0116】まず、ステップS71では、入力するビデオ素材の初期設定を $k=kend$ 、 $j=gop_pcnt$ 、 $fstart=j$ 、 $i_flag=fmode=0$ として行う。そして、ステップS72で、GOPの始まりを検出した時点から、 j を時間的に後ろから見ていき、ステップS74のなだらかに変化するフェードアウト点の検出処理工程と、ステップS75の通常に変化するフェードアウト点の検出処理工程を実行する。そして、これらの検出処理工程を、ステップS76及びステップS77での処理及び判定を介して繰り返す。

【0117】上記ステップS74に示したなだらかなフェードアウト点の検出処理工程を、サブルーチンを示す図11とタイミングチャートを示す図11を用いて説明する。

【0118】ここでも、例えば、DC_LOW=25、DC_HIGH=70、第1のしきい値DELTA1=-5、第2のしきい値DELTA2=10、定数 $k1=0.8$ 、定数 $k2=1.0$ 、RATE_UP=3.0、BR_DELTA=50と設定する。

【0119】図12のタイミングチャートでは、(a)が平均強度のレベルDCの変化を示す。また(b)がGOPのDC値の平均値AVR_DCの変化を示す。この(b)にはDC_LOWと、DC_HIGHとが破線で記されている。また、(c)がGOPのBピクチャとIピクチャの比率 b_rate (図中B/Iと記す)の変化を示す。また、(d)がGOP単位のモード $fmode$ の変化を示す。

【0120】このようなゆっくりと立ち下がるフェードアウトの区間のGOPでは、隣接するGOPとの画像の差が小さいため、上記図12の(c)に示すような b_rate の大きさも急激には変化しない。そこで、このような場合には、上記図12の(c)に示すような b_rate を用いずに、図12の(b)に示すような avr_dc の値の変化だけで上記フェードアウト区間を検出する。

【0121】まず、図11のステップS81では、時間的に後ろのGOPから順にサーチしていき、連続する2つのGOPの avr_dc の値、すなわち現在のGOPの avr_dc

と一つ後ろのGOPの avr_dc が共に、DC_LOW以下であるか否かを判定し、共に以下であればステップS82に進み現在のGOPの $fmode=1$ とし、この位置のGOP番号(j)を $fstart=j$ とする。

【0122】また、ステップS83では1つ後ろのGOPの avr_dc との変化量が予め設定されている第1のしきい値DELTA1以下であるか否かを判定し、以下であればステップS82に進み $fmode=1$ とし、この位置のGOP番号(j)を $fstart=j$ とする。

【0123】また、ステップS84では1つ後ろGOPの avr_dc との変化量が上記第1のしきい値DELTA1と第2のしきい値DELTA2との間にあるか否かを判定し、間にあればステップS85に進み $fmode$ を2とする。

【0124】そうでない場合には、 $fmode=1$ とし、その位置のGOP番号(j)に対して $fstart=j$ とする。

【0125】ステップS86及びステップS87を通して、 $fmode$ が2の時に、 avr_dc がDC_HIGH以上になったか否かを判定し、以上になればステップS88及びステップS89を通して、 $fstart$ からその位置までのGOPの重み係数 $rate_ctl[i]$ を $avr_gop_genbit / gop_genbit[i] * k1$ として求める。重み係数 $rate_ctl[i]$ の大きさはステップS90～ステップS93を介して1.0～RATE_UPの範囲に制限される。

【0126】そして、これらの処理は、ステップS94及びステップS95を通して繰り返され、 j 番目までいったら、ステップS96で $fmode=1$ として、図10のステップS75に進む。

【0127】この図11に示したゆっくりしたフェードアウト点の検出処理では、重み係数 $rate_ctl$ は、初期設定値を1としたときに、ステップS89に示したように、GOP単位のdifficultyと、平均値との比率に定数 $k1$ を乗じた値とする。

【0128】もし、重み係数の値を2倍といった固定値に設定すると、補正されたdifficultyが大きくなりすぎ、不必要な大きさまでビットレートが配分される可能性がある。また、全体のdifficultyの分布によっては、2倍では不十分なこともあり得る。

【0129】そこで、本発明の画像符号化方法によって重み係数を決定することで、補正されて配分されたレートは平均ビットレートの近傍の値になるため、過小、過度にレートが補正されることがなくなる。

【0130】次に、上記図10のステップS75に示した通常に変化するフェードアウト点の検出処理工程を、サブルーチンを示す図13とタイミングチャートを示す図14を用いて説明する。

【0131】ここでも、例えば、DC_LOW=25、DC_HIGH=70、第1のしきい値DELTA1=-5、第2のしきい値DELTA2=10、定数 $k1=0.8$ 、定数 $k2=1.0$ 、RATE_UP=3.0、BR_DELTA=50と設定する。

【0132】図14のタイミングチャートでも、(a)

が平均輝度のレベルDCの変化を示す。また(b)がGOPのDC値の平均値AVR_DCの変化を示す。この(b)にはDC_LOWと、DC_HIGHとが破線で記されている。また、

(c)がGOPのBピクチャとIピクチャの比率b_rate(図中B/Iと記す)の変化を示す。また、(d)がGOP単位のモードfmodeの変化を示す。

【0133】このような通常で立ち下がるフェードアウトの場合には、フェードインの区間の場合と異なり、フェードアウト点では、図14の(c)に示すb_rateの変化量がそれほど大きくなりえないため、フェードインの場合と同じ方法では検出が難しい。

【0134】そこで、まず、図14の(b)に示すavr_dc値と図14の(c)に示すb_rateの変化量からフェードインの位置を検出し、そこから時間的に最も近い位置で、図14の(b)に示すavr_dcの変化量が第2のしきい値DELTA2以上になったところをフェードアウト点としている。

【0135】まず、図13のステップS101で、一つ後ろのGOPのb_rateと現在のGOPのb_rateとの差がBRのしきい値BR_DELTAより大きく、かつ現在のGOPのavr_dcの値がDC_LOW以下であるか否かを判定し、YESのときにはその位置がフェードイン点であるとしてステップS102でi_flag=1とする。

【0136】ステップS103では、i_flag=1である一つ後ろのGOPのavr_dcとの変化量が第2のしきい値DELTA2より大きいのか否かを判定し、YESのときにはステップS104及びステップS105を通して、その位置をフェードアウト点であると判断し、GOPの重み係数rate_ctlを、 $rate_ctl[i] = avr_gop_genbit / gop_genbit[i] * k2$ として求める。

【0137】重み係数の大きさは、ステップS106～ステップS109を通して、1.0～RATE_UPの範囲に制限される。そして、ステップS110及びステップS111の処理及び判定を通して、フェードアウト点から3個前までのGOPの区間の重み係数を計算する。

【0138】このように、図4に示した、ステップS22でのフェードイン点の検出、重み係数の算出処理工程、及びステップS23でのフェードアウト点の検出、重み係数の算出処理工程により、上記図5～図14を参照して説明したようにして、各重み係数が算出された後には、各重み係数を用いてステップS24でdifficultyの重み付けが行われる。

【0139】このdifficultyの重み付け処理工程のサブルーチンを図15に示す。まず、ステップS121で $k=j=0$ としてから、ステップS122でgop_start[k]が1になったか否かを判定してGOPの先頭を検出する。

【0140】そして、ステップS123及びステップS124を通して、仮エンコードによって得られたgop_genbitに上記各重み付け係数rate_ctlを乗算し、重み付けしたdifficultyを算出する。そして、ステップS125

及びステップS126を通して、この算出処理が繰り返され、ステップS127を介したステップS128での判定により最後のフレームまで処理が繰り返されたのが分かると終了する。

【0141】その後、上記図2のステップS16に進み、上記図4のステップS24で得られた補正符号化難易度に応じて、ターゲットビット量を計算し、全体のビットレートを割り当ててビットレートが極端に小さくならないように保護する。

【0142】なお、上記画像符号化方法による圧縮画像信号が記録された記録媒体は、暗い部分を介して明暗が変化する区間を上記フレーム内の平均輝度のレベルの変化に基づいて検出し、上記明暗が変化する区間における上記符号化難易度を補正し、上記補正した符号化難易度に応じて上記割り当てビット量を決定した画像を記録しているので、フェードイン/アウト時でも画質の劣化を生じさせない。

【0143】また、上記画像符号化方法により符号化された画像情報を記録媒体に記録するのではなく、例えば有線回線などの伝送路を用いて伝送する画像伝送方法においては、少なくとも一つのフレーム内符号化画像と、フレーム間順方向予測符号化画像及び双方向予測符号化画像を含んで構成される画像符号化グループ単位での平均輝度の時間変化に基づいて、暗い部分を介して明暗が変化する区間を検出し、この区間における上記符号化難易度を補正し、この補正した符号化難易度に応じて上記割り当てビット量を決定するので、フェードイン/アウト時でも画質の劣化を生じさせないで伝送できる。

【0144】

【発明の効果】本発明に係る画像符号化方法及び装置は、例えばDVD等の圧縮されたビデオ信号をパッケージメディアに蓄積させるエンコードシステムにおいて、エンコード素材からフェードイン/フェードアウト点を検出して重み付けすることで、レート配分時に通常処理よりも多くのレートを割り当て、フェードイン/フェードアウトポイントの画質を改善することができる。

【0145】また、本発明に係る記録媒体は、フェードイン/アウトが行われている画像でも再生時に画質の劣化を感じさせない。

【0146】また、本発明に係る画像伝送方法は、フェードイン/アウトが行われている画像でも受信再生時に画質の劣化を感じさせない。

【図面の簡単な説明】

【図1】本発明に係る画像符号化方法及び装置の実施の形態となるビデオエンコードシステムの具体例のブロック図である。

【図2】上記ビデオエンコードシステムに使われるビデオエンコードコントローラ内部のビット配分計算部の実行するビット配分計算処理プログラムを説明するためのフローチャートである。

【図3】上記実施の形態となるビデオエンコードシステムの全体的な動作を説明するためのタイミングチャートである。

【図4】上記ビデオエンコードシステムにおけるフェードイン／フェードアウト点でのDIFFICULTYの重み付け処理を説明するためのフローチャートである。

【図5】上記ビデオエンコードシステムによるフェードイン点の検出処理工程を説明するためのフローチャートである。

【図6】上記ビデオエンコードシステムによるゆっくりとしたフェードインポイント検出処理工程を説明するためのフローチャートである。

【図7】ゆっくりと立ち上がるフェードイン点の検出処理を説明するためのタイミングチャートである。

【図8】上記ビデオエンコードシステムによる通常のフェードインポイント検出処理工程を説明するためのフローチャートである。

【図9】通常のフェードイン点の検出処理を説明するためのタイミングチャートである。

【図10】上記ビデオエンコードシステムによるフェードアウト点の検出処理工程を示すフローチャートである。

【図11】ゆっくりとしたフェードアウトポイント検出処理工程を説明するためのフローチャートである。

【図12】ゆっくりと立ち下がるフェードアウト点の検出処理工程を説明するためのフローチャートである。

【図13】通常のフェードアウトポイント検出処理を説明するためのフローチャートである。

【図14】通常のフェードアウトポイントの検出処理を説明するためのタイミングチャートである。

【図15】フェードイン／フェードアウト点でのdifficultyの重み付け処理工程を説明するためのフローチャートである。

【図16】従来のビデオエンコードシステムの具体例のブロック図である。

【図17】GOP構造を説明するための図である。

【図18】上記従来のビデオエンコードシステムにおけるエンコード処理を説明するためのフローチャートである。

【図19】上記図18に示したエンコード処理におけるビット配分計算処理を説明するためのフローチャートである。

【図20】シーンチェンジ指定によるピクチャタイプの変更を説明するための図である。

【図21】チャプター指定によるピクチャタイプの変更を説明するための図である。

【図22】GOP単位の評価関数の例を示す特性図である。

【図23】V BVの計算方法を説明するための特性図である。

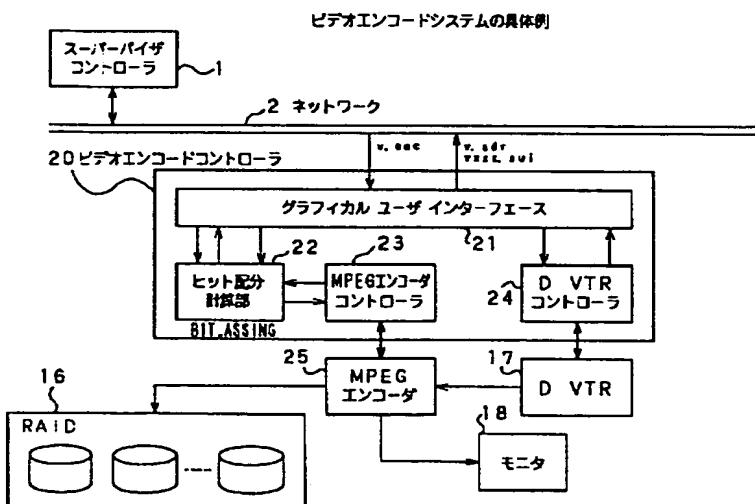
【図24】ターゲットビット配分の具体例を示す図である。

【図25】従来のビデオエンコードシステムによるフェードイン区間でのレート配分処理を説明するためのタイミングチャートである。

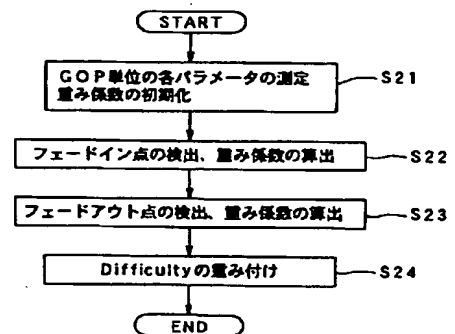
【符号の説明】

1 スーパーバイザコントローラ、20 ビデオエンコードコントローラ、22 ビット配分計算部、23 MPEGエンコーダコントローラ、25 MPEGエンコーダ

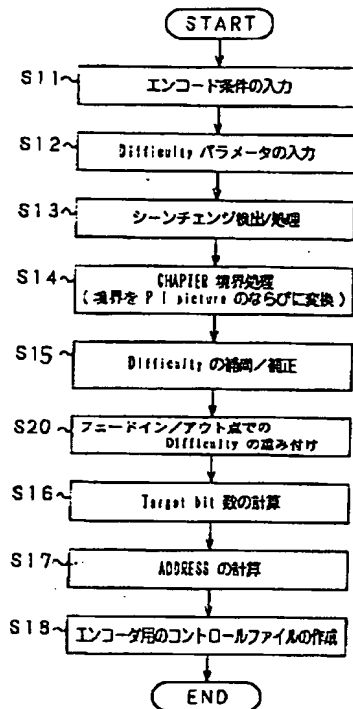
【図1】



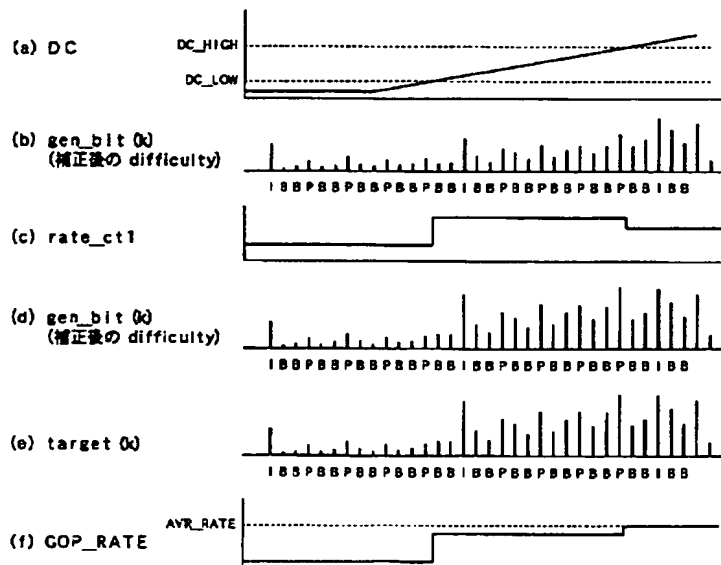
【図4】



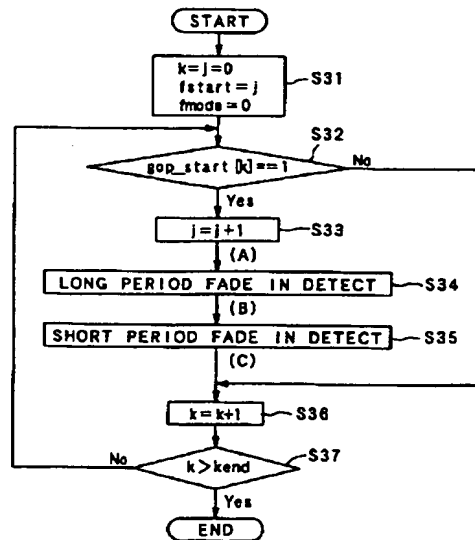
【図2】



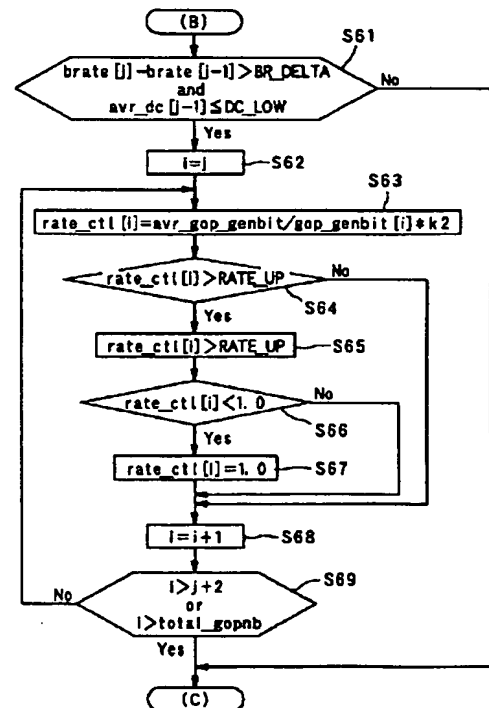
【図3】



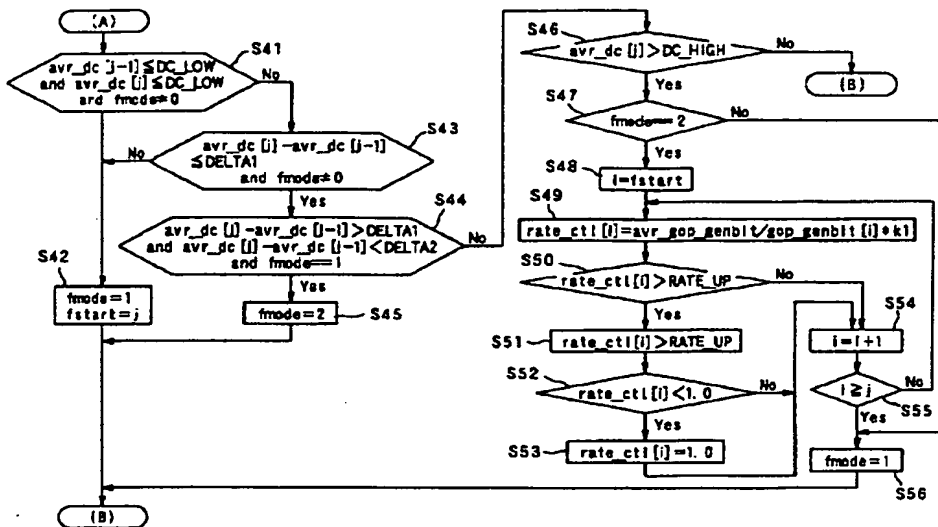
【図5】



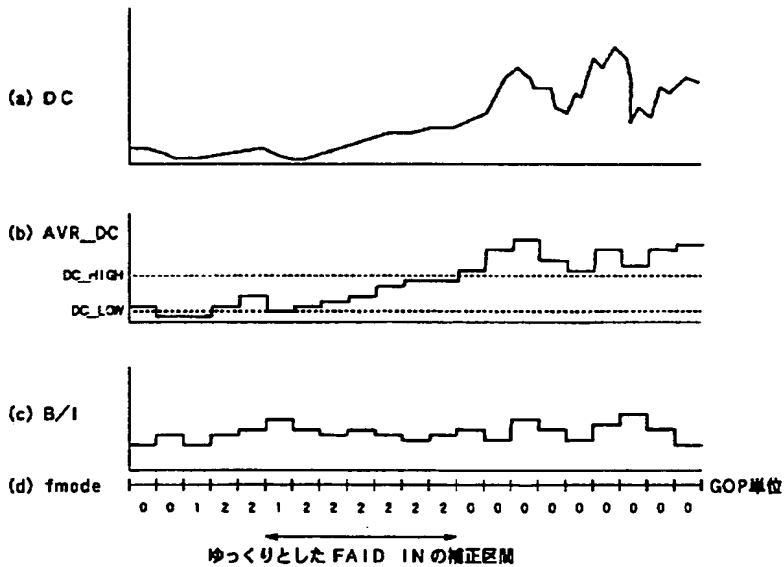
【図8】



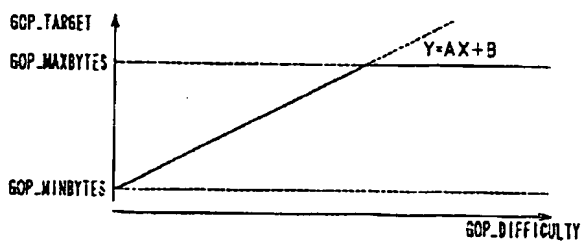
【図6】



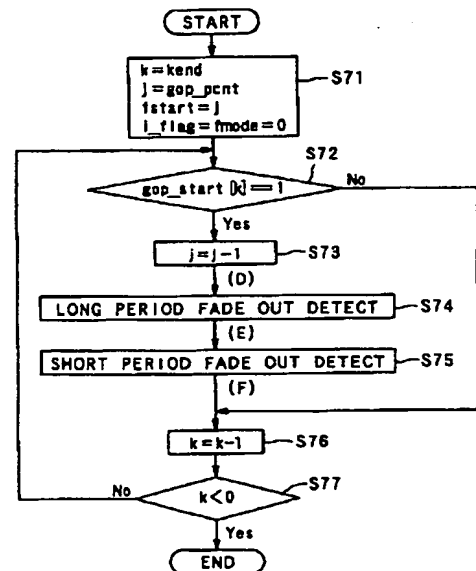
【図7】



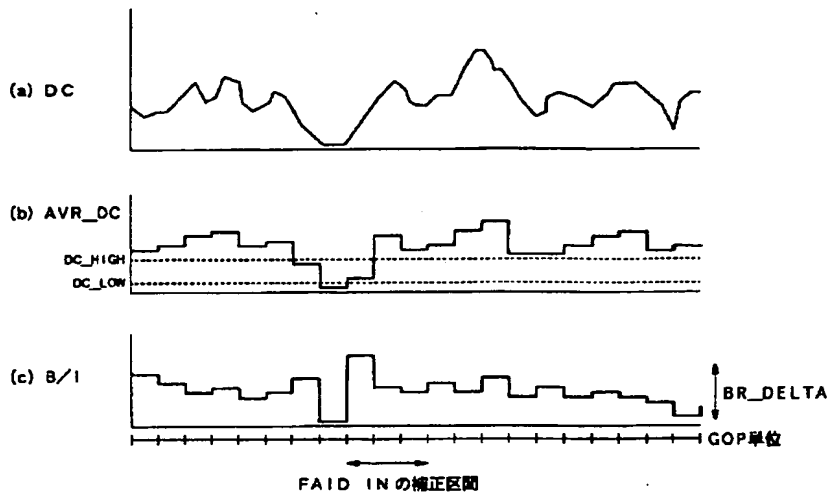
【図22】



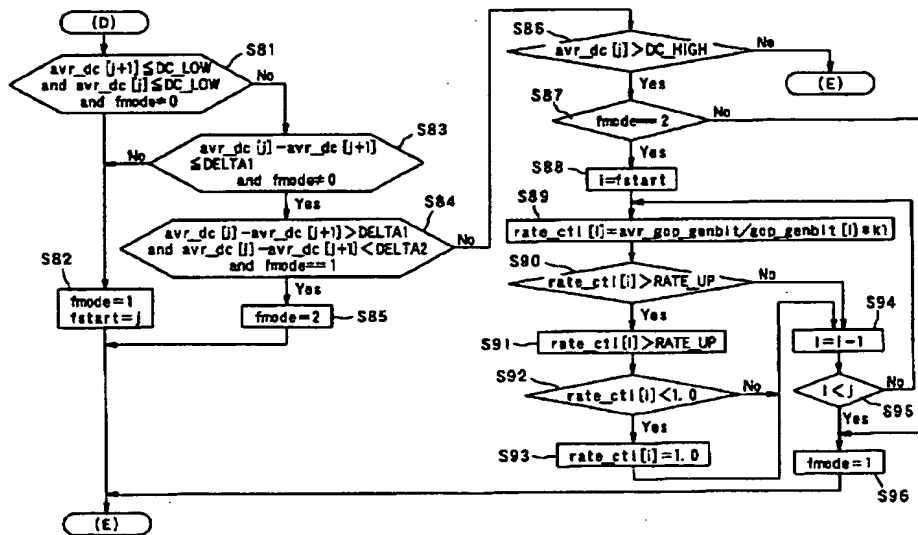
【図10】



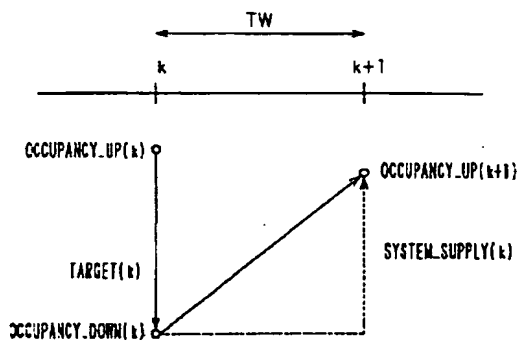
【図9】



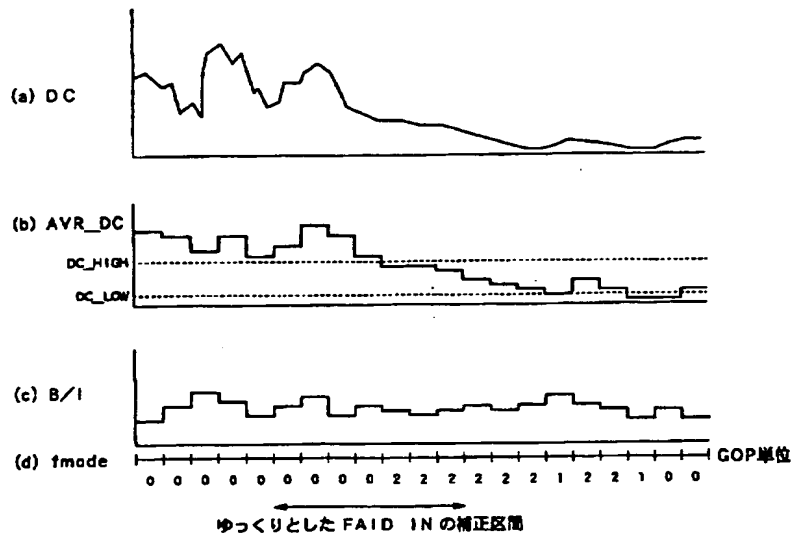
【図11】



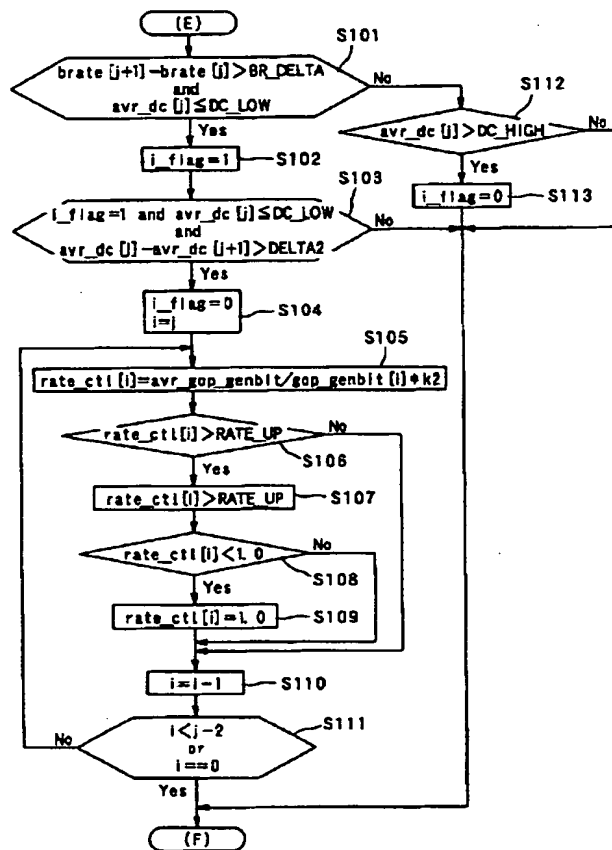
【図23】



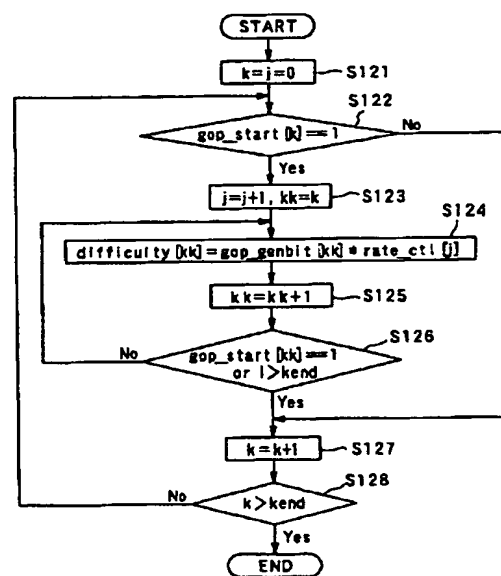
【図12】



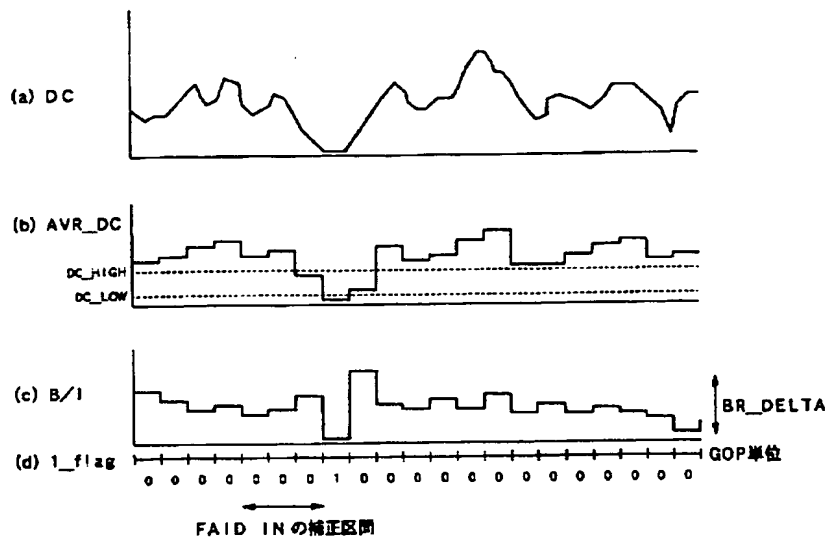
【図13】



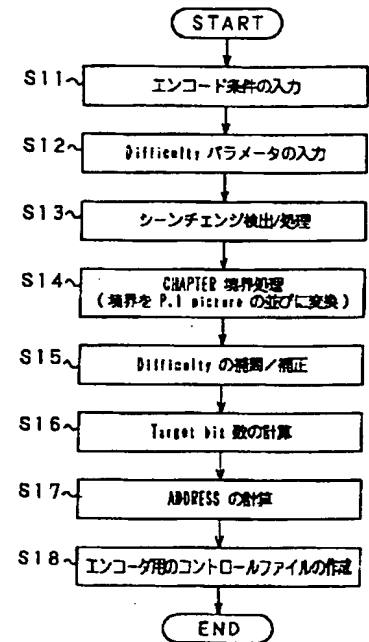
【図15】



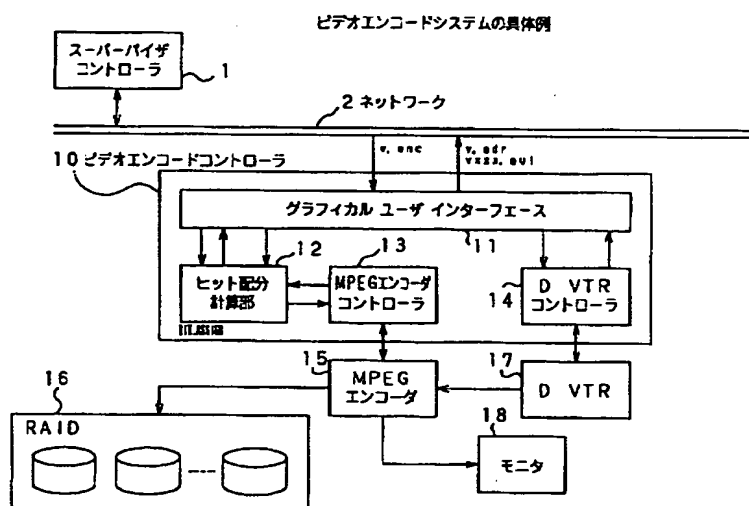
【図14】



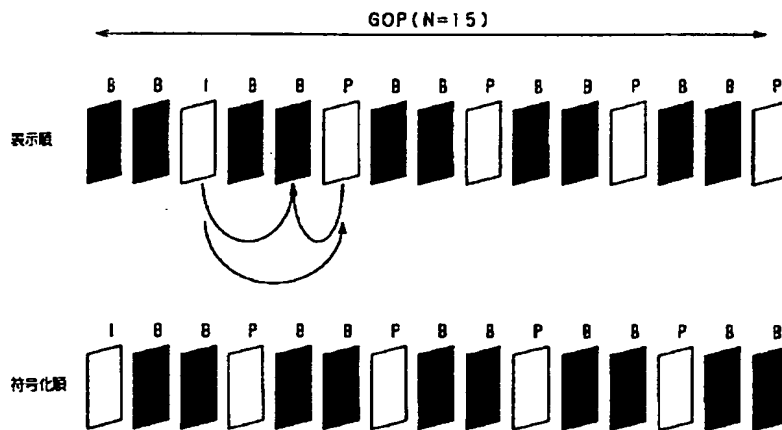
【図19】



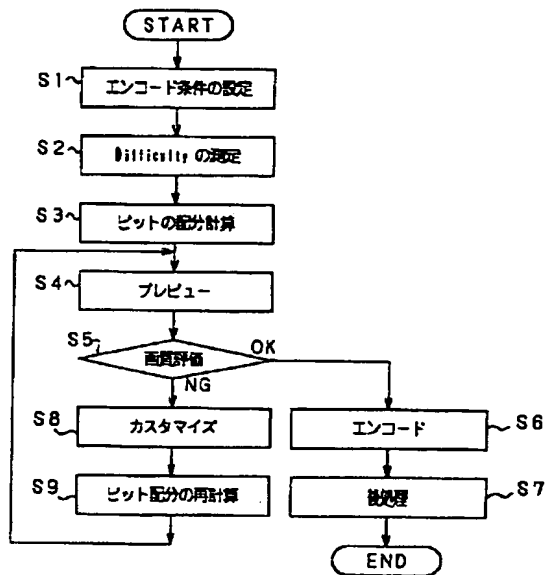
【図16】



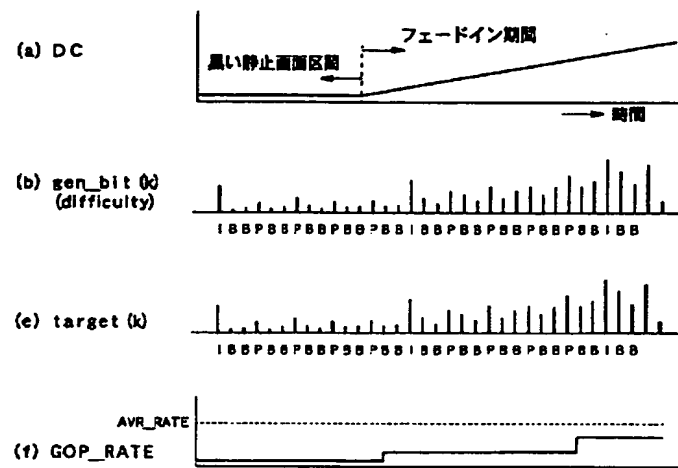
【図 17】



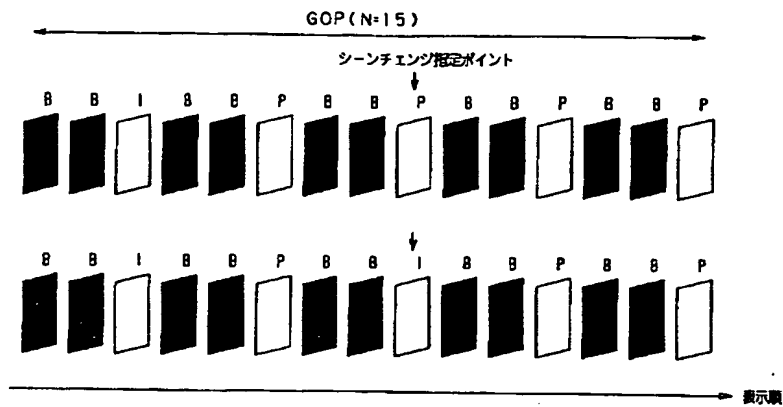
【図 18】



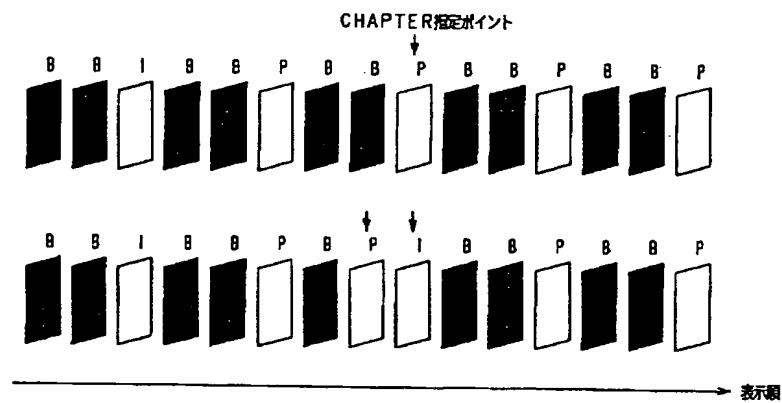
【図 25】



【図20】



【図21】



【図24】

